

一种轻量级架构的 ETL 系统设计与实现

黄怀毅, 杨路明

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

摘要: ETL 系统是构建数据仓库的关键组件。通过数据抽取、转换和装载工作, 完成处理海量数据和构建数据仓库的任务。如何提高数据处理效率和增强系统稳定性是值得研究的问题。根据电信资源数据省级集中分析系统下的实际应用背景, 提出了一种轻量级架构的 ETL 系统, 该架构采取的是“化整为零”的思想, 把原先核心的抽取和转换的任务部分划分到各个相应的子系统中做初步的处理, 这样处理的优点在于应用中减轻了核心 ETL 引擎的承载负担, 各个抽取任务单独进行互不依赖, 提高了抽取效率。通过在具体应用环境下系统运行表明采用该架构 ETL 系统具良好的使用效果。

关键词: ETL; 数据抽取; 数据转换; 数据装载; 架构

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)06-0202-04

Design and Implementation of Lightweight Architecture of ETL System

HUANG Huai-yi, YANG Lu-ming

(School of Information Science & Engineering, Central South University, Changsha 410083, China)

Abstract: ETL system is a key component responsible for constructing data warehouse. It processes large volumes of data and builds data warehouse through data extraction, transformation and loading. How to efficiently shorten the execution time and enhance system stability is a big challenge. According the application of practical background of telecommunication resources analysis system, gives a lightweight architecture of ETL system. The framework adopted the “piecemeal” thinking, the original core of the extraction tasks assign to the various parts of the subsystem to do preliminary processing, with the advantages of this application is to reduce the core ETL engine burden, separate tasks from each other, and improve extraction efficiency. The specific operation proves it to be efficient.

Key words: ETL; data extraction; data transform; data load; architecture

0 引言

ETL 是数据抽取、转换、装载的过程, 它是构建数据仓库的关键步骤。ETL 的本质是把不同数据库中面向业务的数据转换成能够为上层分析应用系统提供决策支持的数据。ETL 过程中各个活动是相互关联顺序执行, 它主要完成以下几类活动^[1]:

①数据抽取: 从不同平台下异构数据源采用不同形式的接口进行数据的原始抽取。这个步骤中是把数据放入临时的中间介等待后续步骤的清洗与转换。为提高抽取效率通常采用数据分割和缩短数据抽取周期的方法。

②数据转换: 在临时中间介中按照预先设定好规则对异构数据库中抽取上来的数据进行清洗、转换、拆分、汇总。同时还包括对数据重新格式化, 完成对缺失

数据的填补, 对数据的完整性验证和对关键数据进行重新构建。这个步骤是获得统一格式、干净数据的保证。提高数据转换的效率是整体 ETL 的数据迁移过程效率提高的关键。

③数据装载: 将临时中间介中转换后的数据加载到数据仓库中。数据加载策略包括全量加载策略和增量加载策略。对于电信企业级应用, 同时采用全量与增量互补的加载策略, 在一定长周期内对数据进行全量加载, 小周期实时性要求较高的情况下采用增量加载。这个步骤是整个 ETL 稳定性的重要指标。

如何提供一个高效的数据抽取、转换、装载的过程和保证过程中系统的稳定性是建设 ETL 系统成败的关键。在自行设计和开发 ETL 过程中一般只着重对三个基本活动的局部进行优化, 对整体架构的考虑没有很好的解决方案。因此, 文中在中国电信资源集中分析平台的实际应用背景下, 提出了一种以减轻 ETL 核心引擎承载能力来提高抽取效率和提高系统稳定性为思想的轻量级的易于实现和部署的 ETL 系统架构。该架构同时引入了 XML 技术、多线程技术和分布式

收稿日期: 2007-09-21

作者简介: 黄怀毅(1982-), 男, 湖南郴州人, 硕士研究生, 主要研究方向为软件架构技术与数据库; 杨路明, 教授, 博士生导师, 主要研究方向为计算机网络与大型数据库技术。

技术,采用这种架构的 ETL 系统在数据量大,业务种类变化快,运行效率,日常开发和维护的方便性应用扩展的灵活性方面得到大大的提高。

1 传统 ETL 体系结构与技术

图 1 是目前流行的 ETL 系统的基本体系结构图。整个 ETL 过程都是由元数据驱动的。虽然整个结构简单,但是由结构图可以看出其缺点也是显而易见的^[2]。首先数据抽取、数据转换和数据加载三类服务的封装是粗粒度的,开发时每块服务的开发量大,开发完成后部署较为复杂,占用的空间大。其次不能很好地协调各个数据源的抽取任务,容易造成抽取总线数据流量太大,抽取引擎负荷较重,直接影响整体 ETL 系统的效率。第三,由于抽取到临时存储区的数据之前未进行任何的验证等操作,虽然是事先有以元数据为驱动的转换规则,但是规则定义复杂,不能实时性动态地进行规则变换,对于脏数据的过滤和数据转换中错误的处理需要额外占用数据转换引擎进行处理,这就容易造成转换引擎的效率变低,而且影响整个数据迁移过程中的系统的稳定性。再次,由于采用的服务封装粒度较粗,数据在加载过程当中如果一旦发现数据质量有问题,很难准确定位出错原因^[3],查找错误较为困难。

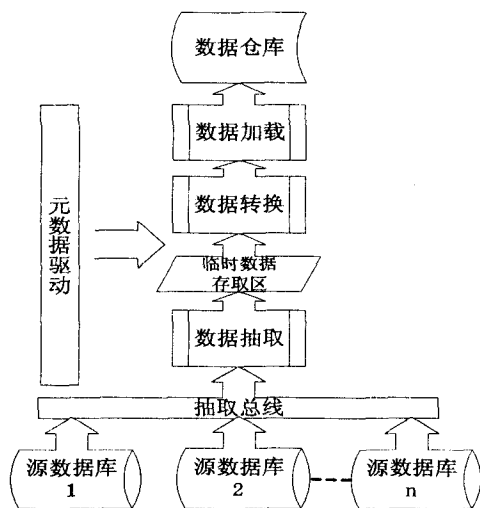


图 1 目前主流 ETL 系统基本架构

从目前流行 ETL 系统对上述几个问题的解决方案来看,目前主要的 ETL 系统基本还是采用的传统的架构方法,服务封装的粒度依然较粗。在优化方面,只是对其中某个问题进行了优化,对整体优化和架构的变化考虑较少。因此,考虑整体优化是提升系统效率与稳定性的目标,通过电信企业级实际应用的不断改进,提出了对 ETL 三个基本服务进行“化整为零”法,分别把三个服务或者过程进一步进行拆分建立一个轻

量级架构的稳定的易于实现和部署的 ETL 系统。

2 改进的轻量级 ETL 架构与实现

2.1 轻量级架构产生背景

根据实际电信资源省级集中分析系统的数据迁移需求,该系统需要采集实施省份下属所有地市级(本地网)电信资源的数据。由于电信资源数据量巨大,再加上各个地市级的数据量差别较大,在实施数据迁移的过程中如果只采用传统架构的 ETL 系统进行单服务器集中部署,会造成数据迁移的时间周期过长,整个系统的稳定性较差,而且服务器要求高,另外采购高档服务器的费用昂贵。通过不断对整个项目数据迁移的要求分析发现,本地网级负责资源数据服务器利用率较低。在充分考虑本地网级数据库服务器额外加载服务对原有系统的影响和服务器最优承载能力,提出了把传统数据抽取、数据转换部分拆分到本地网级服务器进行承载,以小粒度进行数据抽取和转换两类服务的封装,产生一个轻量级架构的 ETL 系统,并通过应用事件收集和任务调度管理进一步减轻核心 ETL 系统的负荷,提升整个系统的效率和稳定性。

2.2 改进的轻量级 ETL 架构

图 2 给出了改进的轻量级架构的 ETL 系统整体结构图^[1,4]。粗箭头代表数据流动的方向,小箭头代表控制流方向。

下面由下至上对整体架构的各个组件进行介绍。

(1)数据库层。这一层在传统的 ETL 系统中只包括了源数据库或者其他存储格式的数据,例如文件等。在改进的架构里面把初步抽取与分析组件放入到了数据层面。首先由应用事件收集器采集应用事件所需要的数据要求,再通过预先定义的规则和动态生成的规则产生新的规则,新规则生成后通过任务调度组件生成对应的 SQL 代码,该 SQL 代码被发布到数据库中,最后初步抽取与分析组件调用相应数据库的企业管理器执行语句并进行初步的主动抽取工作。这样充分利用了数据库服务器的空闲时间,相当于把原有数据抽取和数据转换的工作部分分解,分别承担,实质是把原来的数据抽取大组件分解成了小组件。

(2)ETL 层面^[5]。因为采用的初步数据抽取与分析组件的方案,所以在真正 ETL 服务器上的压力与效率大为提高。在抽取上把抽取组件划分为增量组件和全量组件,抽取时通过系统总线按需求进行数据抽取。增量组件和全量组件可以单独配置单独部署,初步实现组件定制。在数据转换中,把数据转换组件分解成规则组件、转换进程组件、转换拆分与规则反馈组件。其中规则组件是根据预先定义的规则进行转化,这个

组件在对于转换规则较为简单实时性要求不高的环境下配置使用。转换进程组件是一个公共组件,它是实际数据转换的运行部分。转换分析与规则反馈组件与应用事件收集和任务调度相关,当收集到的应用事件对数据的实际要求时可以通过这两个组件分别生成 SQL 代码并且反馈给规则定义器,最后通过任务调度处理组件自动运行。在数据装载中,采用多线程技术,提高数据转载的效率。装载进程可以使用数据库的 API 批量数据导入,每个进程只需提交一次数据库 SQL 语句,就可以导入大量类似结构的数据,数据库上下文不需要切换,同时入库使用的连接池技术节省了数据库连接的时间,大大提高装载的效率。

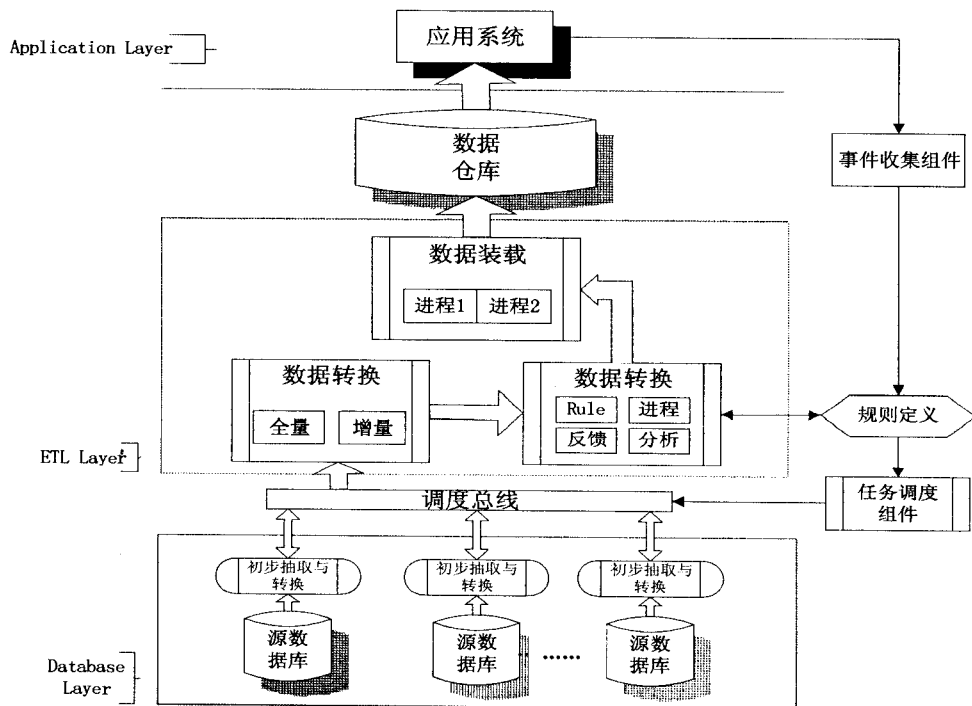


图 2 改进的 ETL 架构图

应用事件收集组件与任务调度组件是整个过程的主轴,是该结构系统下各个组件模块协调的关键。应用系统发出数据要求事件,事件收集组件进行收集并以元数据方式对事件进行描述,描述以 XML 格式通过规则定义器解释并继续传递给数据转换组件中的反馈与分析组件,由此自动生成 SQL 代码,最后任务处理组件通过调度总线调用初步抽取与转换组件自动完成代码的运行。任务调度可以采用定义一张状态表,实时记录各个源数据库服务器的状态,以服务器状态为标志调度各个子抽取作业。科学分配任务运行的时间。另外,任务调度组件可以根据状态信息,迅速定位抽取中的错误的来源帮助找到错误原因。增加事件收集组件和任务调度总线的优点在于减轻中央服务器数据转换的压力,很好地协调了各个活动的进行。

(3)应用层。根据实际电信资源集中分析系统的背景需求,可以留出一个 API 编写一个数据迁移事件发生器,事先定义好数据迁移定义格式。由架构中事件收集器收集数据迁移事件。在对各个组件的封装上来看,采取了各个组件拆分后的小粒度封装,基本上做到了“需要什么,部署什么”的轻量级架构思想,对于实际应用系统不需要的服务不会强制性额外地附带进 ETL 系统中去。

2.3 系统的实现与应用

整个系统采用 JAVA 语言来实现。下面的源数据库包括 Oracle 和 Sybase 两类数据库。数据最后装载的数据仓库采用的是 Oracle 数据库。下面是几个关键组件的实现。

初步抽取与分析组件是整个系统实现轻量级架构的关键。这个组件采用针对不同数据源预定的方法,分别部署到不同的服务器上,组件向上保留 API。组件与系统总线采用 Web Service 的方式进行服务调用,当有新的单个抽取任务时,根据任务的目标源数据库指向与目标源数据库服务器的工作状态,任务调度组件寻找相应的目标源服务器对源数据库进行服务调用,启动相应的抽取

与分析组件进行初步抽取与分析。经过初步分析与抽取的数据暂时保留在下面本地网级的服务器临时数据表里面等待中心 ETL 服务器的抽取。在对服务的调用中,任务管理组件以 XML 字符串的形式传递从其上个步骤生成好的数据抽取与分析规则(其中包括了以文本形式存在的 SQL 代码节点)。在该组件编码过程中发现,由于不需要再考虑复杂的数据映射和转换规则,只需要对传递的 XML 字符串的解析后填充事先定义好的规则模版,代码量迅速减少,缩短了系统开发的时间。

中心 ETL 系统借鉴了开源 ETL 工具 Kettle^[6],在此基础上改造并重新封装。增量抽取组件利用时间戳原理在原 Kettle 上增加代码,实现增量抽取。增量抽取和全量抽取之间可以在系统启动时通过一个开关参

数进行选择性加载。这样可以减少内存占用,启动速度加快。数据转换以元数据为驱动,建立一个记录数据转换规则的元数据库,由于数据转换中部分任务已经转化到了下面各个分服务器承担,所以规则的定义上没有传统的复杂,规则的定义组件单独实现,定义好的规则存储在元数据库中。公共部分的转换组件可以用 Kettle 的转换引擎实现。反馈和分析组件在获得事件收集器的事件之后查找对应的元数据描述,组合成元数据信息集合体和 SQL 代码传递给任务调度组件,任务调度组件再附加服务器状态等信息后再次组合成 XML 字符串,以字符串参数形式调用下层的服务器进行初步抽取和分析。各个数据抽取的小模块都可以实现单独编码单独实现小粒度封装,模块之间通过元数据库为中心进行数据读取传递。

数据装载采用了 JAVA API 1.5 的进程调度包实现。各进程之间整体调度,提高了加载速度。对于应用事件收集组件本系统采用了 Socket 实现,应用事件需预先定义事件传递的格式。任务调度组件中采用线程池和队列技术。结合一个源数据库服务器状态表实时读取与更新状态,动态调度各个子抽取任务的运行。

整个系统经过不断的调试改进已经部署到某省份电信资源集中分析系统中。中心 ETL 系统服务器与资源集中分析系统共用,这样节省了服务器硬件采购的成本。分拆后的初步抽取与分析组件部署到了该省份下属 14 个本地网的资源数据库服务器中。资源集中分析系统所需要的来源于每个本地网的数据平均大小为 2G 左右,每个本地网的数据库服务器:2 * 1.0 GHz CPU, 2 G 内存。中心 ETL 服务器:2.7 GHz CPU, 2 G 内存。图 3 给出了系统开发初期用一般 ETL 工具直接通过 JDBC 抽取下属 14 个数据库数据和文中改进的 ETL 系统抽取下属 14 个本地网数据库数据的时间比较。从图中可以看出,采用文中这种轻量级架构的 ETL 系统的抽取时间比一般架构的 ETL 系统抽取时间少。整个系统的稳定性也相应的得到提高。

3 结束语

在传统 ETL 架构基础上提出的改进轻量级的

ETL 架构提高了数据抽取和数据转换的效率,对数据装载也有改善。这种架构的系统组件划分合适,开发上可实现定制,封装的粒度小,服务配置做到了“需要什么,配置什么”的要求。通过在某省电信资源集中分析系统的实施,有效地解决了海量数据遇到的效率低下的瓶颈问题,同时它配置灵活,中心服务器承载压力减轻,稳定性也相应的增强。但是还存在不足,比如说支持的数据源太少,目前只限于 Oracle 和 Sybase 两类数据源,通用性方面还有待提高。对于任务调度组件的完善考虑的问题也还很多,比如任务调度队列的结构优化等。

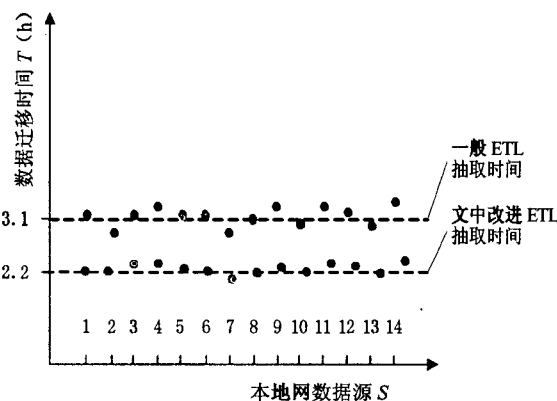


图 3 两种 ETL 系统数据迁移时间对比

参考文献:

- [1] 陈伟江,郭朝珍.分布式 ETL 中协同机制的研究与设计[J].通信学报,2006,27(11):177-181.
- [2] 尤玉林,张宪民.一种可靠的数据仓库中 ETL 策略与架构设计[J].计算机工程与应用,2005(10):172-174.
- [3] 孙伟,张忠能.ETL 架构研究[J].微型电脑应用,2005,21(3):13-15.
- [4] 花海洋,李一凡,赵怀慈.基于分布式数据仓库技术的 ETL 系统的研究与应用[J].微计算机信息,2006,22(10):144-146.
- [5] Vassiliadis P, Simitsis A, Skiadopoulos S. Modeling ETL activities as graphs[C]//In Proc. of 4th International Workshop on the Design and Management of Data Warehouses. Toronto, Canada: [s. n.], 2002:52-61.
- [6] 俞东进,赵明德.一种轻量级数据仓库平台的设计与实现[J].计算机工程,2005,31(15):206-208.

(上接第 201 页)

- Internet Computing, 2000, 4(4):36-45.
- [2] 胡毅时,怀进鹏.基于 Web 服务的单点登录系统的研究与实现[J].北京航空航天大学学报,2004,30(13):236-239.
- [3] 卢清平,杨柳,许晓东.一个基于 Yale-CAS 的单点登录解决方案[J].合肥学院学报:自然科学版,2005,15(3):37

-40.

- [4] 吴洁明,杨轶鑫.基于 Portal 的统一身份认证系统研究与开发[J].航空计算技术,2004,34(4):89-91.
- [5] 吴建武.基于公钥证书的 cookies 安全实现方案[J].微计算机信息,2006,7(3):136-138.